



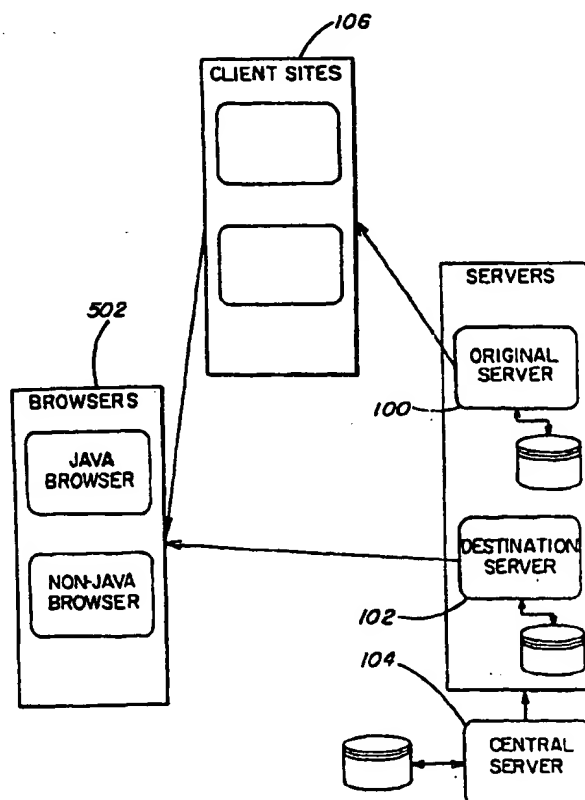
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 17/00		A2	(11) International Publication Number: WO 00/60490
			(43) International Publication Date: 12 October 2000 (12.10.00)
(21) International Application Number: PCT/US00/08784 (22) International Filing Date: 3 April 2000 (03.04.00) (30) Priority Data: 09/285,628 3 April 1999 (03.04.99) US (71) Applicant: MUCHOINFO.COM, INC. [US/US]; 4101 East Louisiana Avenue, Denver, CO 80222 (US). (72) Inventor: WEISER, Nicolas, S.; 772 Race Street, Denver, CO 80206 (US). (74) Agents: PARKS, Jonathan, C. et al.; Kirkpatrick & Lockhart LLP, Henry W. Oliver Building, 535 Smithfield Street, Pittsburgh, PA 15222-2312 (US).		(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>	

(54) Title: ARCHITECTURE FOR AND METHOD OF COLLECTING SURVEY DATA IN A NETWORK ENVIRONMENT

(57) Abstract

A network based survey system utilizing links from client sites to a set of logical servers to establish communication between users of the client sites and the survey system. The system preferably selects a percentage of the available users for participation in the survey using an adaptive selection process which adjusts to the load on the system. Preferably, users are deterministically mapped to a single one of a set of available servers, allowing user profile information to be stored only on a single server. The profile information can be used to allow a user to continue a survey at a later time or to implement question skip patterns or other traditional survey techniques. Alternatively, a central database is provided to collect, store, and process survey results.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Title: Architecture For And Method Of Collecting Survey Data In A Network Environment

Inventor: Weiser, Nicolas S.

Field of the Invention

The present invention relates to the field of administering surveys electronically and
5 more specifically to doing so over a computer network to users of that network.

Background of the Invention

The Internet is becoming a major force both in society in general and in commerce
specifically. It offers unprecedented access to information and, increasingly, to retailers. As
10 use continues to increase, it also represents an enormous population of users about which
very little is known.

As is typical when a major new marketplace opens, businesses associated with the
Internet want information about the user population. What are their interests, what are the
demographics, what can and can't be sold via the Internet? Unfortunately, traditional market
15 or audience survey techniques work poorly in this area. For a physical marketplace, such as a
new shopping mall, the customers can be randomly physically contacted as they arrive or as
they shop. Potential customers can be assumed to come from the surrounding locale, and
randomly contacted by telephone or mail. With the Internet, geographic boundaries are
irrelevant. Of more concern is the ease of finding the site and the language used. This can
20 make it extremely difficult to identify the potential customers of a web site.

While actual visitors to a web site can be contacted as they enter the site, difficulties
still arise. Many of the users of the Internet are anonymous to some degree. Seldom are
actual names used, a login ID being the norm. These are often cryptic, either by choice or
necessity. Some users will take further steps to intentionally hide their identity. Further, a
25 single user may have multiple login IDs.

In this environment, traditional techniques can be difficult to apply. Selection of the
survey respondent to fit certain criteria is difficult where there is no interviewer to see, hear,
or question the respondent. The control provided by the human interviewer in preventing a
single user from responding many times is not available.

The logistics of administering surveys on the Internet can be daunting. Users are in the millions; web sites in the hundreds of thousands and both numbers are growing rapidly. Limiting the web sites to only those relevant to a particular content area may still leave thousands of sites. How can all of the users to all of these sites be contacted? How to select a
5 sub-population of these visitors to provide a statistically valid sample?

Clearly, a survey program could be installed at each of the sites. However this would take resources away from the web site itself, in terms of processing time, storage, and network bandwidth. Few web site owners or administrators would willingly give up such resources. In addition, it would take a significant amount of time and cost to install and test
10 thousands of survey systems. This could easily consume many weeks and tens of thousands of dollars before the first survey could be presented. Similar costs may be incurred to remove the survey program at the completion of the survey.

Compounding this is the pace at which the Internet grows and changes. A period of months can be a very long time in the Internet marketplace. A retailer trying to position a
15 web site, or trying to identify a cause for dropping sales cannot wait months for a survey to be complete. Weeks may even be too long. They may need to see at least preliminary information within days. Traditional techniques can not hope to meet these timelines.

Traditional techniques are also labor intensive. In-person interviews and telephone interviews are slow, one on one, processes. Data entry of the responses can also be labor
20 intensive and error prone. All of these increase the cost of a survey.

While many web site administrators may be interested in performing their own surveys, they likely lack the knowledge, experience, and resources to do so properly. Those who do, may lack the time or interest to develop a valid survey. Further, each web site only has access to the users which visit that site. While it may provide an accurate picture of their
25 population it may not be valid for the general web population. Where multiple sites combine their data for a larger picture, the problem of duplicate responses arises. If the sites are all in the same content area, the likelihood of web users visiting more than one of the sites is very high.

The above and other problems make it difficult to administer a useful, statistically
30 valid survey on the Internet within reasonable time and budget constraints and relatively few such surveys are conducted. However, the information is desperately needed and that need will continue to grow as the Internet user population and web site base grows. If the

techniques could be developed, the Internet user population also provides a rich resource for surveys of non-Internet topics. The instant access to a diverse worldwide population is appealing even to those companies who do not deal directly with the Internet. In this time of a growing global marketplace, access to international users is highly desirable.

- 5 There is a need for a method of conducting surveys on the Internet, or other network, which is statistically valid, addressing the problem of duplicate responses by anonymous users, does not allow the users to self select, and provides the necessary control of quotas. The system should be able to adapt to variations in such factors as the number of visitors and the completion percentage of users presented a survey. Ideally, this adaptation would occur
- 10 continuously during the administration of the survey, maintaining smooth progress towards the survey goals. The survey method should be scaleable from a single site to integrating the responses of thousands, or even hundreds of thousands, of sites. At the same time, it should be possible to quickly define and initiate the survey without significant impact on the web sites from which the sample population is drawn. Upon completion of the survey, it should
- 15 be possible to easily disconnect the survey from those web sites. When drawing from multiple sites, the system should be able to eliminate duplicate users. Ideally, the system should allow a user to partially complete a survey during one session and return to complete it at a later time, even if connecting from a different site. Traditional techniques such as question skip patterns should also be supported. The survey results should be available
- 20 promptly upon completion of the survey and the availability of intermediate results would be highly desirable. It should be possible to configure the system to either focus on specific user groups or like web sites or to sample a diverse population across the Internet.

Summary of the Invention

The present invention is directed to an apparatus and method for surveying users of a network, such as the Internet, which makes use of links from the pages of independent content providers to provide the initial contact with the system. A set of logical servers
5 administers surveys to a subset of the available users by selecting them randomly using an adaptive process which continuously adjusts the percentage of users selected in response to the number of users visiting the client sites and the percentage of those who are presented a survey that actually complete it.

According to the invention there are provided plural logical servers connected to a
10 network which also serves one or more content provider sites. A link is embedded in a page of the provider sites which connects to a logical server. When a link is activated, communications is established between the user and the survey system. The user is then presented a survey to be completed.

According to an aspect of the invention there may be both original servers which
15 handle the initial communication with and identification of the user and destination servers to which the user is then connected to complete the survey. Preferably, the user is mapped to a destination server by a process which results in the same user always being connected to the same destination server without regard to the client site or original server used to initially access the system. The original and destination servers may be roles played by a single type
20 of logical server which can serve as either type for different connections or at different times.

According to another aspect of the invention a central database server may be provided which collects, stores, and processes the survey results and makes them available to customers of the survey system.

Further in accordance with the invention user profile information may be maintained
25 by the destination server(s) which includes information about the user which allows completion of a survey at a later time, implementation of question skip patterns, and other techniques to improve the validity of the survey.

Still further in accordance with the invention, the survey system may restrict access to only a percentage of those users visiting the client site and may adaptively adjust this
30 response in response to factors such as the number of hits on the client sites and the percentage of users completing a survey which is presented to them.

The advantages of such a system and method are that the population of Web users who visit a site or collection of sites can be surveyed in a statistically valid and timely manner at a minimum cost. A survey can be quickly defined, set up, administered, and the results obtained with little impact on the participating client sites. The system is self regulating in
5 terms of sampling quotas and adjusts to load changes to maintain desired sampling levels. The system can be quickly reconfigured to focus on the visitors to a single site, a collection of sites in a single category, or to a diverse set of sites representative of the Web as a whole. Survey results are available shortly after completion of the survey in a variety of formats including electronically downloadable.

10 The above and other features and advantages of the present invention will become more clear from the detailed description of a specific illustrative embodiment thereof, presented below in conjunction with the accompanying drawings.

Brief Description of the Drawings

- FIG. 1 - provides a block diagram of the Internet.
- FIG. 2 - illustrates an abstract HTML web page.
- FIG. 3 - illustrates a hypertext link from one web page to another.
- 5 FIG. 4 - provides a high level block diagram of the inventive system architecture.
- FIG. 5 - illustrates the creation of links to the original and destination servers.
- FIG. 6 - illustrates the sequence of messages involved in establishing a connection to the system and completing the survey.
- FIG. 7 - provides a block diagram of the major components of a logical server.
- 10 FIG. 8 - graphically represents the selection of participating users from the available population and the reduced number who complete the survey.
- FIG. 9 - is a flowchart of the sampling adaptation process.
- FIGs. 10 A & B - are a data flow diagram illustrating the determination of new sampling parameters.
- 15 FIGs. 11 A - K - are pseudo code of an illustrative implementation of the sampling adaptation process.
- FIG. 12 - illustrates the logical loop arrangement of servers used to collect statistics.
- FIG. 13 - illustrates the merging of template and form to create a survey questionnaire.

Detailed Description of the Invention

The following discussion focuses on the preferred embodiment of the invention, in which the disclosed architecture is used in conjunction with the Internet to collect survey information. However, as will be recognized by those skilled in the art, the disclosed method and apparatus are applicable to a wide variety of situations in which the collection of statistical information using a network is desired.

The following is a brief glossary of terms used herein. The supplied definitions are applicable throughout this specification and the claims unless the term is clearly used in another manner.

Applet - a special form of computer program designed to be downloaded from a host in conjunction with a web page. Typically written in the JAVA language, an applet is unique in that it can be executed on any hardware platform which includes a JAVA engine. This differs significantly from normal programs which are built for a specific computer. Applets are usually restricted in the access which they are allowed to the resources of the computer on which they are executed and the type of network communications they are allowed to perform.

Browser Software - generally a computer program executing on the users local computer which is designed to navigate and display (browse) WWW documents but which includes any software program which provide an interface between a computer network and a user of that network. Examples include NCSA's Mosaic, Netscape's Navigator and Microsoft's Internet Explorer.

Central Server - in the present invention, a data storage server which collects, stores, and merges survey results.

CGI (Common Gateway Interface) - a protocol for how a web server communicates with another program executing on the same computer. Any program can be a CGI application if it handles input and output according to the CGI standard. CGI applications differ from applets in that they run on a specific server and must be compatible with the hardware and operating system provided by that server.

Client Site - in the present invention, a client site is a content provider which has been modified to provide a link to the inventive system. The user first makes contact with the survey system through a link incorporated into one or more pages of the client site.

Destination Server - in the present invention, the server which handles the communication with the user to present and collect the survey information. After the identity of the user is established, the Original Server passes the user on to the Destination Server to handle the remainder of the transaction.

5 Form - in the present invention, a document which contains the survey text in a generic format. This text will be merged with a Template to create the final survey document presented to the user.

HTML (Hypertext Markup Language) - a hypertext document specification language used primarily for the creation of WWW documents. It is a block oriented language which
10 utilizes tags to define formats and features which are then interpreted by browser software.

Hypertext - a method of constructing documents such that there are multiple pathways through the contents that the user can select and follow, rather than only providing sequential access from beginning to end. The pathways are provided by hypertext links which can lead to other documents, other sections of the same document, or to alternate views. The link,
15 (sometimes referred to as a hyperlink) is often embedded in the text of the document and distinguished by the use of a different color, font, style, or any combination of these. This type of link is typically activated by the user selecting, or clicking on, the link. Links may also be hidden from the user and activated automatically by the browser.

ISP (Internet Service Provider) - a company which provides its clients with a presence
20 on the Internet. This may include hosting of the client's web pages and/or access to the Internet via either a dial-up or dedicated connection. An ISP which provides only access may be referred to as an Internet Access Provider (IAP).

Load - generally the utilization of a resource on a computer. It may be expressed either as an absolute number, such as the number of users connected, or as a percentage, such
25 as the ration of the portion of the disk or CPU capacity being used to that available.

Look and Feel - a broad term encompassing most, if not all, aesthetic and some functional elements of how a computer program interacts with the user of that program. This includes, but is not limited to, color and font choices, the types of interactive controls used, and the general layout of visual elements on a screen or page.

30 Original Server - in the present invention, the server with which the user makes initial contact as the result of a link from the client site. The original server establishes the identity

of the user and then passes the user on to the Destination Server to handle the remainder of the transaction.

Server - generally, a computer or program, in a distributed environment which provides a specialized service such as data storage, printing, or communications. In the Internet environment the service is more likely to be specific to supporting the Internet such as a Web server that provides WWW pages to a browser program or a Domain Name Server (DNS) that translates logical network names into numeric addresses.

Template - in the present invention, a document which captures the appearance of the pages on a client site and which includes one or more tokens which will be replaced by the text of the survey.

Transaction Monitor - in the present invention, an application running on a server which gathers the statistics used by the adaptive sampling algorithms.

User - the human user of a computer or software program. With respect to the Internet, the person who is using a browser to access the web. In the present invention, the user is that subset of Internet users who are interacting with the inventive system in some way. The same term is often used to refer to the browser software being used by the human user. Often, the distinction between the user and the user's browser is not important.

URL (Uniform Resource Locator) - one form of a logical link which specifies the location of an object on the Internet, such as a file or another web page. URLs are commonly embedded in HTML web pages to specify the target of a hypertext link. A URL consists of multiple fields containing information about the target of the link. This information includes the access method, or format, of the target; the address of the server on which the target resides; and the path to the target in the server's file system. Other information may be included as necessary.

Web Page - a logical page of HTML text which forms the basic medium of the World Wide Web (WWW) protocol. The page can also include images, sounds, embedded programs (such as applets), and other data types.

WWW (World Wide Web) - a particular protocol used for the Internet and intranets which specifies a graphical, hypertext format which provides a point and click interface to distributed documents via browser software. Often, that portion of the Internet which supports the WWW protocol is loosely referred to as the "Web."

Preferred Embodiment

The disclosed invention is described below with reference to the accompanying figures in which like reference numbers designate like parts.

Internet Overview

5 As the preferred embodiment of the present invention is of use primarily in the context of the Internet, that environment will be briefly described. However, the present invention is not restricted to the present day Internet. It is equally applicable to other network architectures, both present and future. The network provides a communications medium between the various components of the system and any network or communications grid
10 which serves this role is considered equivalent. An abstract representation of the Internet architecture is shown in FIG. 1. While the Internet itself, 500, is often thought of, and even described as a single "backbone" to which all of the systems are connected, this is incorrect. The Internet is a computer network which, by design, has no centralized control. It is a loose agglomeration of a very large number of computers and sub-networks which cooperate to
15 provide the services which are viewed as the Internet. Its model of distributed control sometimes borders on anarchy. No single entity, computer, or communications link is critical to the Internet. Services are duplicated, data storage is mirrored, and communications paths are redundant. This results in a system which is very resistant to failures, or attack, but which can be daunting to use.

20 The cooperation of the various systems involved in the Internet is regulated by a set of communications protocols and interface standards which simplify the system interactions. Chief among these are the protocols and standards which comprise the World Wide Web (WWW or Web). The WWW is a subset of the Internet which provides a user friendly, largely graphical, point and click view of the Web. Access to the Web is typically through
25 the use of browser software. This is a computer program which resides on the Web user's local computer and which interprets and presents information received from the Web. It understands the relevant protocols and assists the user in navigating the Web. It is also instrumental in directing requests to the various servers (such as search engines) and displaying the results.

30 The web user's computer, or more specifically the browser software executing on it, 502 and 504, may be connected to the Internet via a dedicated connection, 502, more common in the workplace, or may connect as-needed via a dial-up connection, 504, through

an ISP, 506. For most purposes these connections are equivalent, differing only in speed or bandwidth and will generally be referred to throughout the specification as a browser, 502. Further the concept of computer is very broad with respect to the user. Any device which provides network browsing capability is contemplated including laptops, personal
5 information managers, and even telephones.

Various types of servers, 508, are accessible via the Web. Most common, of course, are WWW servers which provide web pages conforming to the Web protocols. Many of these WWW servers are what are referred to as "content providers." These are servers which provide information or data (the "content") which is of interest to some or all of the Web
10 users. Other servers include search engines, which help search for content providers of interest, and providers of various support services, such as dynamic name translation, needed by the infrastructure of the Internet itself.

FIG. 2 provides an illustration of a generic web page as displayed by a browser. The browser will typically present the web page information, 511, in a large window, and will
15 provide it's own information in a separate area, 510. The browsers information area often includes a title bar, 514, a set of menus, 516, and a set of buttons, 518. The menus and buttons provide access to the commands supported by the browser. A status line, 520, is also typically provided for the display of messages to the user. Within the web page area a variety of material may be presented to the user. This includes text, 522, image, 524, and graphical,
20 526, information. The types of information which can be presented are expanding rapidly and include sound and full motion video. Within the text area certain segments of the text, 530, may be designated as hypertext links. In a similar manner, the photos, graphics, and embedded controls, such as buttons, 532, can also be used as links.

The hypertext links are one of the chief mechanisms used in navigating the Web.
25 Web pages are almost universally written in HTML which provides for both formatting of the page content and the specification of links to other pages. From within a browser, the user can easily elect to follow any link presented in the page or can choose to continue with the present page. This process is shown in FIG. 3. If a link, 536, from the page currently being displayed, 534, is followed, the browser, 502, then loads a new page, 538, which may be on
30 the same server, 540, as the present page or may be on another server, 542, anywhere on the Web. Following links, a user can easily retrieve pages from dozens of servers scattered

around the globe in just a matter of minutes. Web pages may also contain links which are automatically activated by the browser when a page is loaded.

Supplementing the HTML pages are a variety of scripts and programs which provide more tailored and powerful services. While HTML is a very flexible language for the presentation of documents, it is limited in the complexity and power of the tasks it can perform. When a more involved manipulation is required, such as retrieving data from a database or generating moving graphics, a program will be activated by the browser, often as a result of a link. These programs may be interpreted scripts executed by an extension to the browser, small applications, often called applets, downloaded to the user's computer and then executed, or larger programs executed on remote server which then supplies its results to the browser for display.

Architecture

It is within the Internet environment that the preferred form of the present invention is preferably used. Making use of the distributed server concept of the Web and interfacing with Web servers, the present system surveys users of the Internet, adapts to the use patterns of those users, merges the results, and presents statistically reliable information for use by its clients.

The general architecture of the present invention is shown in FIG. 4. The major components of the survey system are the original, 100, and destination, 102, servers and the central server, 104. A lesser role is served by the client sites, 106, which are modified to link to the system. While the browser software, 502, is significant in that it provides the user with access to the system and presents surveys to the user, it is not part of the inventive system.

The client sites, 106, are generally content providers as they normally exist in the Web. One or more pages on each of the relevant client sites has been modified with a link to an original server, 100. This link may be either a visible link, which the user elects to follow, or may be a hidden link activated by the browser. For the purposes of the present invention, either type will work. The sole purpose of the link is to establish the initial communications between the user and a pre-selected original server. The selection of server is made by the address specified in the link.

Note that in the following discussion of original and destination servers, each of these is a logical server. One or more logical servers may be hosted by a single physical computer. As the load on the servers varies, the logical servers can be moved between computers. This will be entirely transparent to the users, and to the system itself, as it deals with the logical
5 entities. This flexibility also allows the system to adapt to hardware failures by moving the servers off of the faulty computer.

Referring to FIGs. 5 and 6 the process of establishing communication between the user and the inventive system will be outlined. FIG. 5 provides a graphical depiction of the connections and FIG. 6 captures the sequence of messages which occur. As discussed above,
10 the user, 140 in FIG. 5, first requests a page from a client site, message 110 in FIG. 6, and is provided with that page, message 112. The user then clicks on the link to the survey system (or it is automatically activated) resulting in a request to the original server, message 114. This establishes connection, 142 in FIG. 5, to a server, 144, which acts as an original server.

Alternatively, an applet could be associated with the page. The applet could contain
15 one or more links or could retrieve link information such as from a logical server. The applet would then select one of the links and activate it, establishing the connection. From the users perspective, this approach would be equivalent to the selection and use of the single embedded link, and likely indistinguishable. From the system side, the use of the applet provides more flexibility by making available more decision making capability prior to
20 establishing the link to the system. This enables such capability as randomizing the selection of the original server; connecting directly to the destination server as described below; or merely storing the current set of links in a centralized location to remove the dependency of the client sites on a specific set of link addresses. These and other means of selecting the initial link are interchangeable with respect to their ability to determine the link itself and
25 activate the connection.

The first page presented by the original server, message 116, will ask the user for identifying information such as name, age, and birth date. Additional, or alternative, fields could be used as necessary to uniquely identify each user. Clearly social security numbers or drivers license numbers could be used where appropriate. Each of the original servers is
30 provided with a copy of an identical algorithm which maps a user to a single specific destination server, using the data supplied by the user, message 118, in response to the first page. In the preferred embodiment, this is implemented as a hashing algorithm where each

resultant hash code matches the address of a destination server. This address is provided to the user in a second page, message 120, which also contains privacy notices, links to survey rules, and such other "housekeeping" information as may be desirable to present to the user. Completing this screen, message 122, links the user, connection 148 FIG. 5, to the selected destination server, 150, and starts the actual survey process. In alternative embodiments, the original server could cause a page to be transferred from the destination server to the user's browser without first providing the second, housekeeping page. In a further alternative, the mapping algorithm could be implemented in either an applet which executes locally on the users computer, or in a CGI, or other, program executing on the original server. Either of these approaches would also eliminate the second page. These approaches reduce the number of preliminary pages seen by the user, but incur a cost in requiring multiple implementation of the mapping algorithm, one for each type of computer to be supported. The use of Java as the implementation language could alleviate some or all of these problems. The core component of all of these approaches is that, based on the user provided information, each user is always mapped to the same, specific destination server.

Once the user is connected to a destination server, 150, that server has access to local storage, 152, and can retrieve, messages 124 and 126 in FIG. 6, containing information about the user including a history of previous responses. In this way, the destination server can provide survey pages, message 128, which are tailored to the specific user. The simplest application of this is to begin the survey at the point where the user previously left off. This option is available even if the user connects from a different local computer or through a different client site. When the user completes the survey, message 130, the destination server stores the survey results and updates the user's profile, message 132. If the user responds to the first page and then declines to complete the survey when presented the second page, the database may be updated by a message from the destination server to the database, in place of message 124, and the sequence will terminate.

The ability to track users greatly increases the reliability of the information obtained by the survey system. The approach of mapping users to destination servers also decreases the storage needs for the servers. Without this approach, every server would need to have a duplicate copy of every user's information, or there would have to be a single centralized database which all servers would access for the user information. Either approach would incur a severe performance penalty. With the present architecture, a user's information need

only be stored in a single location and the load can be adjusted by varying both the number of logical servers and the number of physical computers on which they are hosted.

In the preferred embodiment, which utilizes a hashing algorithm, a certain amount of growth can be handled by increasing the number of computers being used and distributing the logical servers across these computers. This will work until each server is hosted on its own computer. Up to this point, these changes can be made with little impact on the system as it continues to run. After this point, if the load continues to increase, the number of logical servers will have to be increased. This will require an increase in the number of "bins" used by the hashing algorithm and distribution of a new hashing algorithm to the original servers. The user data will also have to be re-distributed across the new set of logical servers to match the mapping of the new algorithm. This would require a short down time for the system as the changes are implemented.

FIG. 5 also illustrates the connection of a second user, 154, to the system. In this case, server 150, which was the destination server for the first user, acts as the original server in response to the initial connection, 156, and then directs the user to another server, 160, as the destination server. Note that while different users may be mapped to different servers, as illustrated, the same user will always be connected to the same destination server, as discussed above.

It should be noted that in the preceding discussion the "original server" and "destination server" are roles played by the pool of logical servers, 146 in FIG. 5, within the system. Any server can serve as both an original server and a destination server. Alternatively, these roles could be separated and be supported by distinct logical servers.

The central server, 104 in FIG. 4, serves as the central database for the survey system. The database provides a central location for compiling survey responses and generating results. Any of the analysis techniques or tools well known in the art can be applied to the responses once they are gathered together in the database. The results can then be supplied to the survey customer in any desired form: printed hardcopy, magnetic media, electronic download, etc. At the option of the survey system administrator, a variety of data can be supplied, from the raw responses to the final statistical analysis. Because of the computerized, networked architecture of the system, these responses can be made available almost immediately upon completion of the survey. Alternatively, intermediate results can also be provided as they are compiled by the system. This fast response time is one of the

benefits of an on-line survey system which can not be provided by traditional in-person, mail or telephone interview approaches. The central server is a different logical entity than the other servers in the system. If desired, it can be co-located on a physical computer which also hosts one or more of the original, 100, or destination, 102, servers. In the preferred
5 embodiment, the central server is hosted on a separate physical server. This provides an additional level of security because it makes it more difficult to locate, and allows flexibility in terms of shutting down the central server or disconnecting it from the network, whether for maintenance or security reasons. Performance can also be an issue where significant analysis of the results is performed on the central server. This processing would potentially be slowed
10 by the server(s) responding to survey requests, and, in turn, could slow the server responses.

In the present architecture, the survey system is connected to a large number of client sites, but remains largely independent of them. All of the processing associated with gathering and compiling survey data is performed on the original and destination servers and on the central server. The servers also store all of the forms, survey results, and user profiles.
15 A first advantage of this approach is that the system has very little impact on the client sites. Only the inclusion of a link on the client sites web page is required. This makes the inventive survey system more attractive to the client sites because they will incur no performance or storage cost by allowing the connection. A second advantage is that the servers remain completely under the control of the owner of the survey system. Logistics are greatly
20 simplified because there is no need to consult with the client sites prior to making a system change. Decisions as to increasing or decreasing the number of logical or physical servers can be based solely on performance and cost factors affecting the survey system. Security issues are more easily handled since there is no need to share the server systems, which contain the most sensitive data, with other users such as the client sites. In the preferred
25 embodiment, the various servers are not encrypted. However, in an alternative embodiment, the central server would be encrypted to allow access only by the owner and the destination servers would implement an encryption scheme which would allow access only by those users who access the systems through the normal login path via the original server. This would preferably be implemented via a two key encryption system in which the users key is
30 derived from the hash key used in the user to destination server mapping as described above.

Note also that this implementation enables literally unlimited client sites. Because the initial link is created and stored on the client site, the number of such sites has no impact on

the survey system itself. The more critical factor is the number, or more accurately, the frequency with which the links are activated and the surveys presented. The survey system is capable of handling millions of client sites which experience only light traffic as easily as it handles hundreds of sites with very heavy traffic. This range of application greatly increases the utility of the system.

The high level architecture of a logical server is presented in FIG. 7. The majority of the processing is performed by the transaction monitor, 136, which receives all incoming messages. This includes satisfying requests for survey pages, storing survey results when the user completes a survey, and compiling the local site statistics, 138, as a survey progresses.

The transaction monitor also handles the compilation of the system wide statistics and the adaptive processing as described below. Where the user's browser, 244, is Java enabled, it communicates directly with the transaction monitor via the Internet, 210. If the browser, 242, does not handle Java, the use of a local applet is not available. In this case, a CGI script, 134, or program will execute on the server as a front end to the transaction monitor to handle the details of communication with the browser. In this way, the transaction monitor need only support a single interface. The transaction monitor maintains the user data, 140, which includes identifying information as well as historical information, such as which surveys, or portions of surveys, the user has responded to. This allows the user to complete a survey at a later time and enables such survey techniques as question skip patterns. The survey data, 142, includes both the forms and templates used to present the survey and the survey responses.

Features and Functionality

Within the above architecture, each of the logical servers operates with a fairly high degree of autonomy while presenting surveys. At the start of a survey, each server is configured with a set of control parameters. As the survey progresses, each logical server updates its local statistics and the logical servers periodically communicate to update global statistics. The control parameters, local statistics and global statistics are then used by an adaptive selection algorithm, replicated on each server, to actively control the sampling process in order to achieve the goals of the survey. The efforts of the logical servers combine to present a single survey to a distributed user population in a statistically valid manner.

The overall goal is to randomly select from among the available user population a subset to whom surveys will be presented and to then collect and compile the survey results

from those who complete the survey. This process is presented graphically in FIG. 8. Users are initially represented as "hits" on a client site. A hit is essentially one occurrence of a user requesting a page from a server. The hits of concern to the survey system, 144, are those requests for pages which have had a link to the survey system embedded in them. When such a page is requested, a set of code associated with the link executes to determine whether to present a survey to this user. The details of this process are described below. Of those users presented with a survey, 146, some will decline to participate, some will start the survey but not finish, and some will complete the survey. The completed surveys, 148, are collected and compiled to generate the survey results. Note that this process is not self selecting. The visitors to a client site can not participate in the survey unless invited to do so by the system. This helps avoid the bias associated with convenience sampling and redundant user entries, for example.

The decision process as to which user hits to select for survey participation is handled by the original servers. A variety of appropriate techniques are well known in the field of statistical surveys. In the preferred embodiment, the system identifies a fixed ratio of initial hits to completed surveys and then attempts to accomplish that goal. Two simple approaches are available for reaching that goal. Assume that N out of 100 hits will need to be presented a survey in order to meet the completion percentage. The first approach is to present a survey to every Nth hit reported to an original server (several client sites may report hits to the same original server). A second approach is to use a probabilistic or pseudo random process which selects N/100 of the hits with a reasonably random distribution over the visits.

The sampling process periodically adjusts the parameters of the user selection decision process in an attempt to keep the number of survey completions at or near the goal for the survey. While each original server may use a different set of tailored parameters, the same algorithm is used by each server. One purpose of the use of different parameter by each server is to allow a server which handles smaller sites to sample a larger percentage of the users while a server handling a very large site samples a smaller percentage. This would keep the absolute numbers in more equal proportion.

The primary output of the adaptation algorithm is the number of surveys to be presented during the next period. In generating this value, the algorithm also updates predictions for the number of hits expected during the period and the probability that the

presented surveys will be completed. The algorithm is depicted graphically in FIGs. 9 and 10 and presented as pseudo-code in FIGs. 11A & B.

The flow chart of FIG. 9 provides a high level view of the sequence of events. The processing for each iteration occurs at or near the end of the period. Two sets of data are gathered independently. The branch, 162-166, waits for time $(i * \Delta t - T_i)$, where $(i * \Delta t)$ represent the end of period "i" and T_i represents the mean time to complete a survey. At this time system wide (global) statistics are gathered on the number of hits occurring during the latest period and the number surveys completed during the period. Since any survey started after this point would not (on average) be completed within the period, it can be ignored.

Branch, 168-170, waits for $(i * \Delta t)$, the actual end of the period, and gathers the number of actual completions system wide. In the preferred embodiment, these collections are asynchronous and can be carried on concurrently. Upon completion of both gathering steps, the process synchronizes, 172, and begins the adaptation calculations based on these statistics. First, the number of Survey desired to be collected by the end of the upcoming (next) interval is determined, 174. From this, the number of desired collections during the next period can be determined, 176. The estimated probability of completion for the next period, 178, is combined with the desired number of completions to calculate the number of surveys that will have to be presented during the next period to achieve the goal for completions, 180. This value is provided to the survey processing portion of the system, 182, and the next period starts. This process continues iteratively until the desired number of collections have been made, 184.

This adaptive process periodically adjusts the survey system parameters, primarily, to allow for variations in the number of hits on the client pages, and the probability of completion of those surveys presented. In this way the system actively works to achieve its goal in the estimated amount of time and smoothes out variations in collection performance to maintain an even distribution across the duration of the survey. Other adaptive processes, considering other parameters are clearly possible. Using the present process, any form of adaptation used in conventional surveys can be used in a distributed environment. If desired, the frequency of the local adjustments can be higher than that of the collection of global statistics. For example: if the global statistics are compiled every 30 minutes, the servers can adjust their collection parameters at 15 minute or 10 minute intervals if desired.

In the preferred embodiment of the survey system, the above adaptation process makes use of both global and local statistics in adapting to changes. This allows individual servers to adapt to local changes and allows them to utilize individualized control parameters, such as the percentage of hits to which to present a survey. The process is illustrated in more detail by the data flow diagram of FIGs. 10A & B. Further details are available by reference to FIGs. 11A - K. Referring to FIG. 10a, the steps of the adaptation process are illustrated starting at the end of the period and assuming that all system wide statistics have been gathered and stored. Details of this process are discussed below. First, the number of hits to be experienced by the system during the next period are calculated, 188, by retrieving the number of hits during the last period from the stored global statistics, 186, and applying a predictive algorithm. This result is combined with the cumulative total number of hits received by the end of the last period to estimate the cumulative hits by the end of the next period, 190. This result is then multiplied by the constant completion ratio (determined at the start of the survey), available from the survey control parameters, 192, to calculate the desired number of completions by the end of the next period, 194. Subtracting out the number of actual completions achieved by the end of the last period provides the number of completions needed during the next period, 196, in order to achieve the goal for the end of the period. This value is then divided among the available servers to determine the number of completions each servers must achieve, 198. In the preferred embodiment, this series of calculations utilizes global data but is separately calculate by each server, using a common algorithm. In an alternative embodiment this calculation could be performed once, at least to the point of calculating the system wide desired calculations, and the result distributed.

Referring to FIG. 10B, the remainder of the adaptation process is illustrated. This portion of the process starts with the number of desired completions which has been allocated to a particular server and utilizes local statistics and control parameters to develop the parameters for the presentation of surveys during the next period. A calculation of a local probability of completion, 204, generates a value which is combined with the allocated number of desired completions to generate the number of surveys which must be presented for this server to achieve its goal, 206. This value is used during the next period to determine to which users surveys are presented, 208. During the process of presenting and collecting surveys, local statistics on the number of hits, presented surveys, and completions are maintained in local storage, 200. Note that this process has been presented as two separate

logical steps for illustration purposes only. The process can, and preferably is, implemented as a single combined process.

Referring to FIG. 12, the process of gathering the system wide statistics is illustrated. For purposes of statistics collection, the original servers, 210-216, are logically organized as a ring. When it is time to gather the statistics, a designated one of the servers, 210, retrieves its local use statistics and sends them to the next server, 212, in the ring. This server adds in its local statistics and forwards the message to the next server, 214. This process continues until the last server in the ring, 216, forwards the message to the first server, 210. At this point, the completion of the first circuit, the message contains the totals for all statistics being gathered. The message is then forwarded again, following the same path, to allow each server to make a copy of the total for its own use in its adaptation calculations. In the preferred embodiment the number of hits, h , and surveys presented, s , are compiled with one pair of message circuits and the number of collections, c , is compiled with a second independent sequence. In alternative embodiments other statistics can be compiled in the same manner and using any desired number of message circuit pairs. Where an error occurs, such as the failure of a host computer, interrupting the transmission of the statistics, any of several well known recovery techniques can be used, including a time-out followed by retransmission, skipping a server, or reversing the direction of flow.

It is important to note that this ring arrangement is only a logical arrangement of the servers and has no impact on their physical connections or on any other logical arrangement. In an alternative embodiment of the system a different type of logical arrangement could be used. One such anticipated alternative is a binary tree structure which would provide the well known $O(\log_2(n))$ performance improvement over the $O(n)$ performance of the ring structure described above.

In an alternative embodiment, the original servers are not informed of survey completion. This information is maintained on the destination servers. The collection of system side statistics then incorporates both original and destination servers to generate a complete set of statistics.

The JAVA code in FIG. 11 is presented as pseudo code to illustrate a particular implementation of segments of the adaptive process. Other, more complex, implementations are anticipated. As an example, the estimation routines illustrated utilize a simple weighted

average approach. The use of more accurate approaches, such least squares fit, will be used in alternative embodiments.

One feature of the present system is that the survey forms which are used to interface with the users will have a look and feel similar to that of the pages from which they entered the system. The same user, entering the system from two different client sites, would see the same content, but the presentation style would differ. This is accomplished by using a combination of forms and templates as illustrated in FIG. 13. The templates, 218, are essentially web pages created in the style of the client site. They may use colors, icons, background images, etc. which are available from the client site. Embedded in the template are one or more tokens which identify the location at which to insert survey text. In the preferred embodiment, these templates are created manually. In an alternative embodiment, they could be created automatically by scanning the client web page and extracting design elements. The forms, 220, contain the text which comprises the survey itself. The text is separated into one or more sections which correspond to the tokens embedded in the templates. When a survey page is requested, a script, 222, or program running on the server retrieves the template corresponding to the client site associated with the requesting user, and the form appropriate to the user (dependent on the survey to which responding and historical data such as how much of the survey has been completed) and merges them to create the survey page, 224, which will be presented to the user. The merge process includes the steps of replacing the tokens embedded in the template with the corresponding sections of text from the form. In this manner, a single set of survey forms can be presented to the users of many client sites, in a familiar style. Having only a single set of forms significantly reduces the overhead of creating and maintaining the forms. Utilizing the style of the client site makes the survey look more integrated and more acceptable to both the user and the owner of the client site.

In the preferred embodiment, the forms are created by the administrators of the survey system, based on the requirements of the person, or organization, requesting the survey (the survey customer). In an alternative embodiment, the survey customer could create their own forms and templates, possibly with the assistance of a menu driven interface. When authorized by the survey system, the customer could create and update the forms as desired. This ability would allow them to adapt the survey to the responses being received, changing information needs, or other factors important to them. With a knowledgeable customer, there

would be no need for human intervention by the survey system administrator. In a further alternative, survey results could be compiled automatically, or on request of the survey customer, and provided to the customer directly from the survey system.

When these alternatives are combined, a customer could develop the survey, supply the forms and templates to the survey system, provide the needed configuration values (such as sample size desired, skip patterns to use, number and type of sites to survey, etc.), activate the survey collection process, and retrieve the survey results without involving a human from the survey system administrator. This approach would provide significant benefits in terms of turn-around time and responsiveness for the customers.

The design of the inventive survey system is such that it offers several advantages for collecting survey data. As discussed above, rapid turn-around of results is possible. The architecture is scaleable from a single host with a single logical server to a large number of hosts and logical servers. This allows the system to be adapted to surveying the users of a single site or the users of thousands of sites. Where a web site owner desires audience research focused on that site, the system can be configured with link only from that site, with the option of differentiating different pages or entry points into the web site. The results will consist solely of user responses originating from that site. The system can also be configured to sample a large number of sites, all related to a particular content area (such as snow sports, or gardening) for audience research of that content area with a diverse sample population. The system can, of course, also connect to diverse types of web sites to collect information on the general Web population. Additional capability can be enabled by recording with each response the user and the client site through which they entered. This would allow post processing of the responses to extract data specific to a single site or content area.

While the preferred form of the invention has been disclosed above, alternative methods of practicing the invention are readily apparent to the skilled practitioner. The above description of the preferred embodiment is intended to be illustrative only and not to limit the scope of the invention.

Claims

I/We claim:

1. Where one or more users are accessing a network via respective browser software, the network including one or more content providers, a system for gathering survey
5 information comprising:

one or more logical servers connected to the network and deployed independently of

operational infrastructure for each of said one or more content providers,

wherein at least one of said one or more logical servers stores one or more

survey questionnaires; and

10 an interface on a page on at least one of said one or more content providers, wherein

the interface connects said respective browser software to one of said one or

more logical servers when said respective browser software accesses said

interface, thereby allowing said respective browser software to communicate

with said one of said one or more logical servers over the network.
- 15 2. The survey system of claim 1 wherein said logical servers comprise at least one original

server and one destination server.
3. The survey system of claim 2 wherein said interface includes a link that connects to said

original server, and wherein said one or more survey questionnaires are stored on said

destination server.
- 20 4. The survey system of claim 3 wherein said original server and said destination server are

each able to provide the services of the other.
5. The survey system of claim 1 further comprising means for selecting a specific one of

said one or more logical servers to provide at least one of said one or more survey

questionnaires to a corresponding one of said one or more users.

6. The survey system of claim 5 wherein said means for selecting includes a computer program transmitted by said one of said one or more logical servers over the network to the respective browser software for said corresponding one of said one or more users for execution.
- 5 7. The survey system of claim 5 wherein said means for selecting is responsive to information provided by the corresponding one of said one or more users.
8. The survey system of claim 7 wherein said means for selecting deterministically selects said specific one of said one or more logical servers for all transactions with said corresponding one of said one or more users.
- 10 9. The survey system of claim 1 further comprising a central database connected to the network and logically distinct from said one or more logical servers, wherein the central database comprising non-volatile storage for survey results transmitted from said one or more logical servers over the network.
10. The survey system of claim 1 wherein at least one of said one or more logical servers
15 comprises non-volatile storage for profile information about at least one of the one or more users.
11. The survey system of claim 10 wherein said profile information for a specific one of said one or more users is stored on only one of said one or more logical servers.
12. The survey system of claim 10 wherein said profile information about a user comprises
20 data specifying which of said one or more questionnaires is to be presented to the user.
13. The survey system of claim 1 wherein said survey system restricts access to said one or more logical servers to a percentage of those of said one or more users who access said page on said at least one of said one or more content providers, and wherein said survey

system further comprises means to adaptively adjust said percentage while a survey is being presented.

14. The survey system of claim 13 wherein said adaptive adjustment means is responsive to a load on more than one of said one or more logical servers.
- 5 15. The survey system of claim 13 wherein said adaptive adjustment means adjusts said percentage for each of said one or more logical servers individually and is responsive to at least one of the following:
- a first performance value which is specific for each individual server in
 - said one or more logical servers, and
 - 10 a second performance value encompassing more than one of said one or more logical servers.
16. The survey system of claim 1 wherein said one or more survey questionnaires comprise at least one template specifying an aesthetic element and at least one form comprising one or more questions to be presented to a user from said one or more users and wherein said
- 15 template and said form are combined when presented to said user.
17. The survey system of claim 1 wherein said survey system is used by customers of the survey system to administer one or more surveys to the one or more users and wherein said survey system further comprises means for on-demand creation of said one or more surveys by said customers.
- 20 18. Where one or more users are accessing a network via respective browser software, the network including one or more content providers at least one of which makes available one or more pages for access by the respective browser software, a method of surveying the one or more users comprising:
- establishing a connection between the respective browser software and a first logical

server when a link to said first logical server is activated, wherein at least one first logical server is provided to store and present one or more survey questionnaires, wherein each first logical server is deployed independently of operational infrastructure for each of said one or more content providers, and
5 wherein said link is provided on at least one of said one or more pages; and said first logical server transmitting at least one of said one or more survey questionnaires to the respective browser software over the network in response to the activation of said link.

19. The method of surveying of claim 18 further comprising providing a second logical server
10 that is deployed independently of operational infrastructure for each of said one or more content providers, wherein said link connects to said second logical server when activated.

20. The method of surveying of claim 18 wherein a plurality of said one or more pages contains corresponding links and wherein said method further comprises:
15 identifying a user from said one or more users prior to establishing said connection; and
deterministically connecting the user to the same first logical server independently of which of said corresponding links was initially activated.

21. The method of surveying of claim 18 further comprising selecting, from among a portion
20 of said one or more users activating links available on said one or more pages, a percentage of said portion of said one or more users to be connected to said first logical server with essentially equal probability of connection for each user in said portion of said one or more users.

22. The method of surveying of claim 21 wherein selecting periodically adapts to a load on said first logical server by adjusting the percentage of said portion of said one or more users.
23. The method of surveying of claim 18 further comprising uniquely identifying each of said one or more users connected to said first logical server so as to continue presentation of a survey to said each of said one or more users at a corresponding point where said each of said one or more users previously stopped.
24. The method of surveying of claim 18 wherein said method further comprises uniquely identifying each of said one or more users connected to said first logical server so as to implement a survey technique requiring selection of a subset of questions from said at least one of said one or more survey questionnaires for presentation to the each of said one or more users.
25. The method of surveying of claim 18, further comprising providing means to select said link only in those of said one or more pages that are associated with a common category of goods or services.
26. The method of surveying of claim 18 further comprising providing intermediate survey results while survey administration continues.
27. The method of surveying of claim 18 further comprising providing survey results in an electronic format over the network.
28. Where one or more users are accessing a network via respective browser software, the network including one or more content providers at least one of which makes available one or more pages for access by the respective browser software, wherein at least one of said one or more pages contains a link to a survey system, said survey system comprising: at least one original logical server connected to the network and serving as a target of

the link;

at least one destination logical server comprising non-volatile storage for one or more survey questionnaires and profile information about at least one of the one or more users, wherein said profile information for a specific one of said one or more users is stored on only one destination logical server, and wherein said one or more survey questionnaires comprising at least one template specifying an aesthetic element and at least one form comprising one or more questions to be presented to a user from said one or more users and wherein said template and said form are combined when presented to said user;

means for selecting a specific destination logical server to provide said one or more survey questionnaires to the user, wherein said selecting means deterministically selects the same destination logical server for all transactions with said user, said selecting means being responsive to an activation of the link;

means for restricting access to said specific destination logical server to a percentage of those of said one or more users who access a link-containing page in said one or more pages; and

means to adaptively adjust said percentage while a survey is being presented, wherein said adaptive adjustment means is responsive to a load on more than one of said original and destination logical servers.

FIG. 1

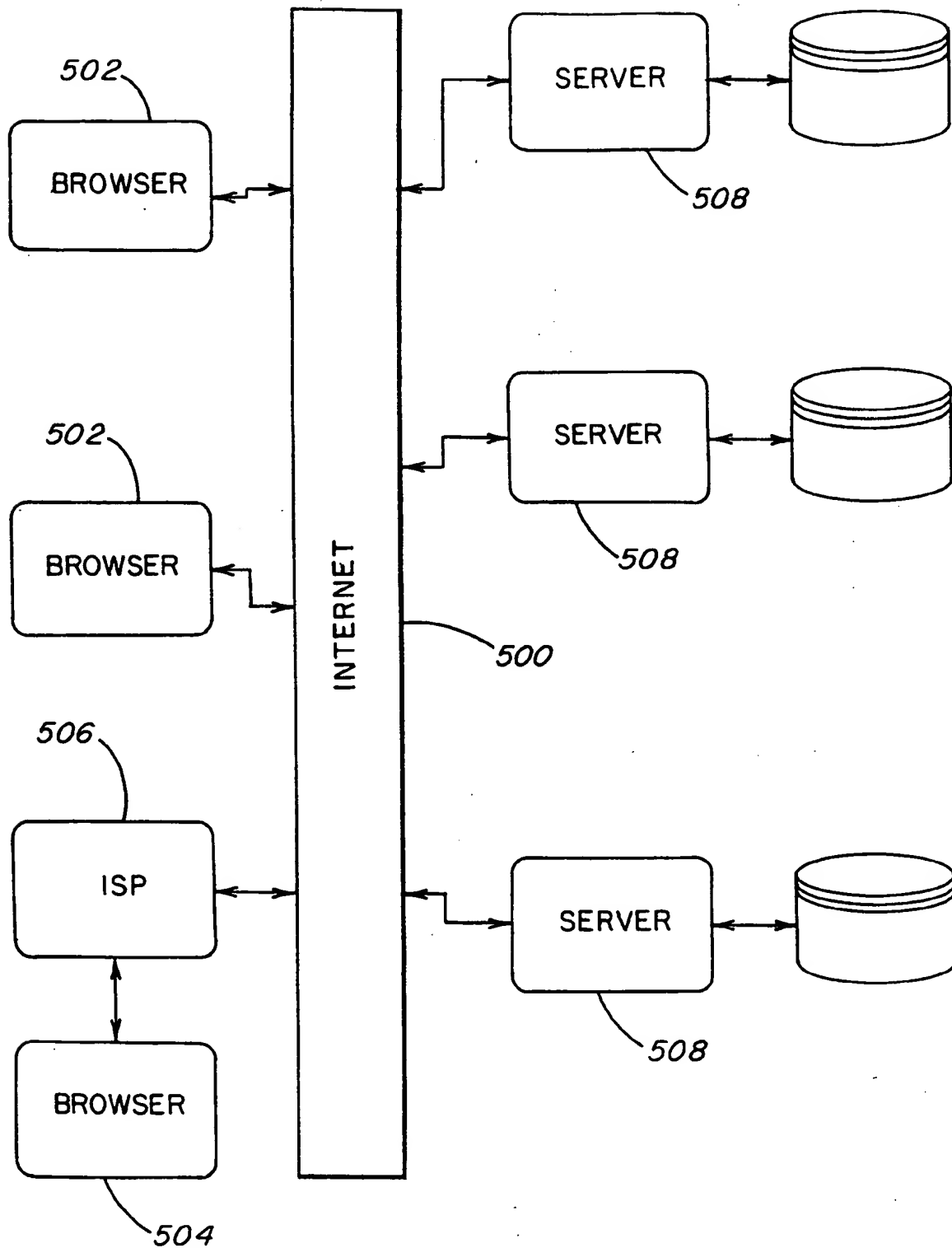


FIG. 2

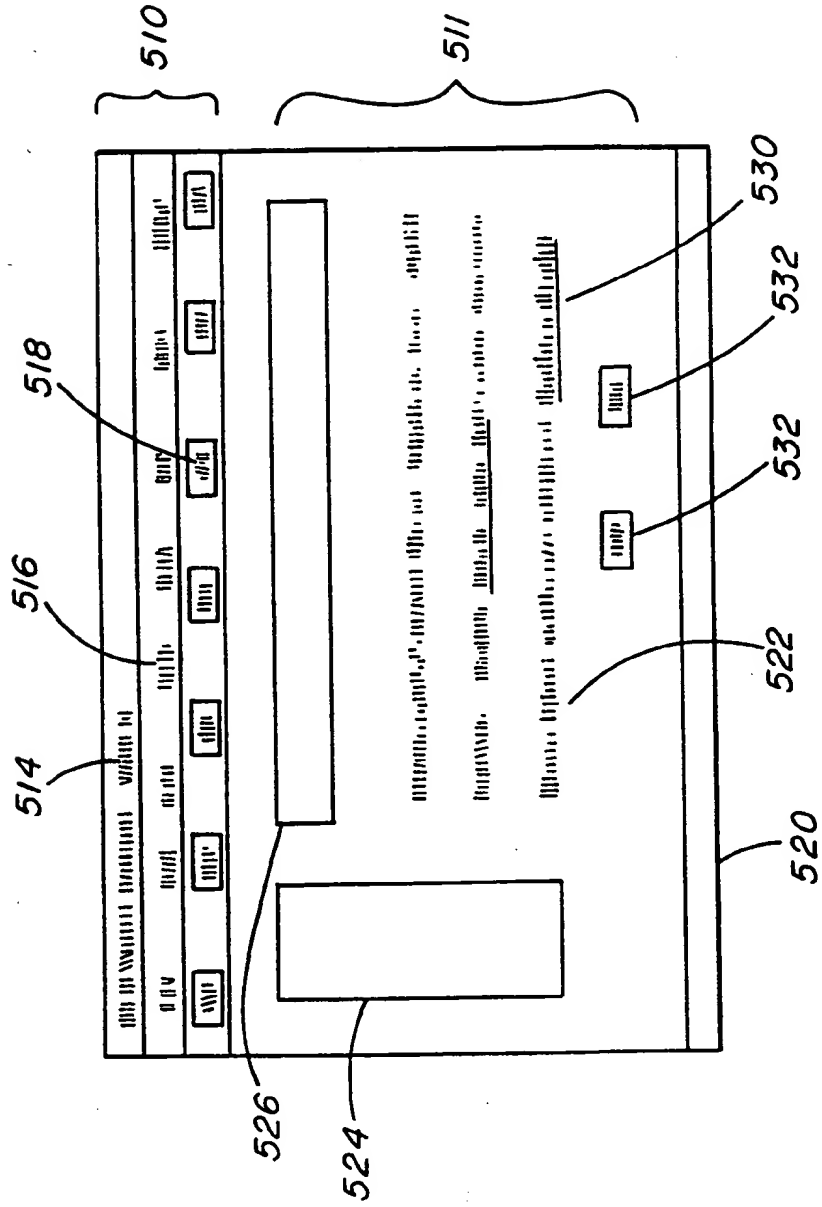


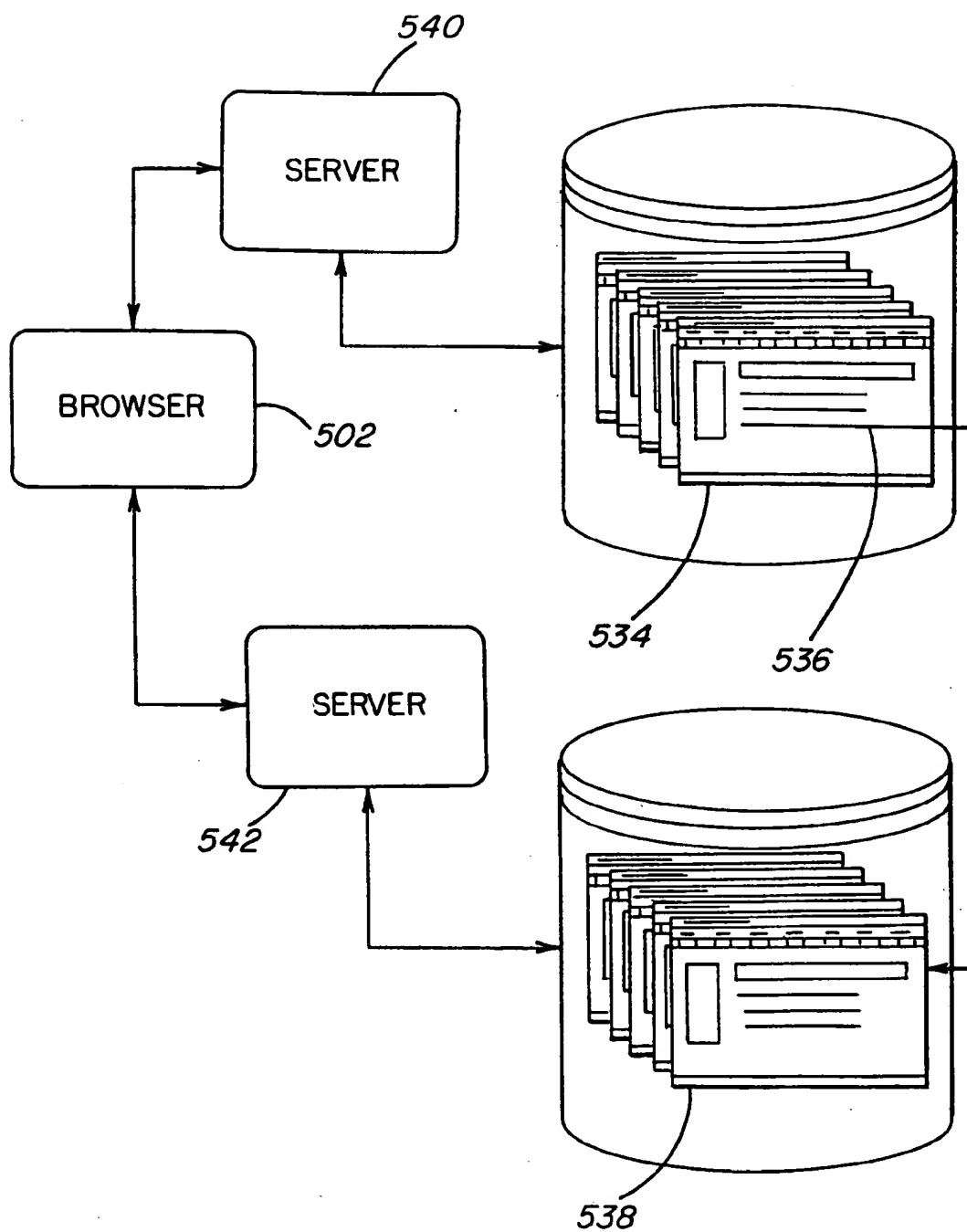
FIG. 3

FIG. 4

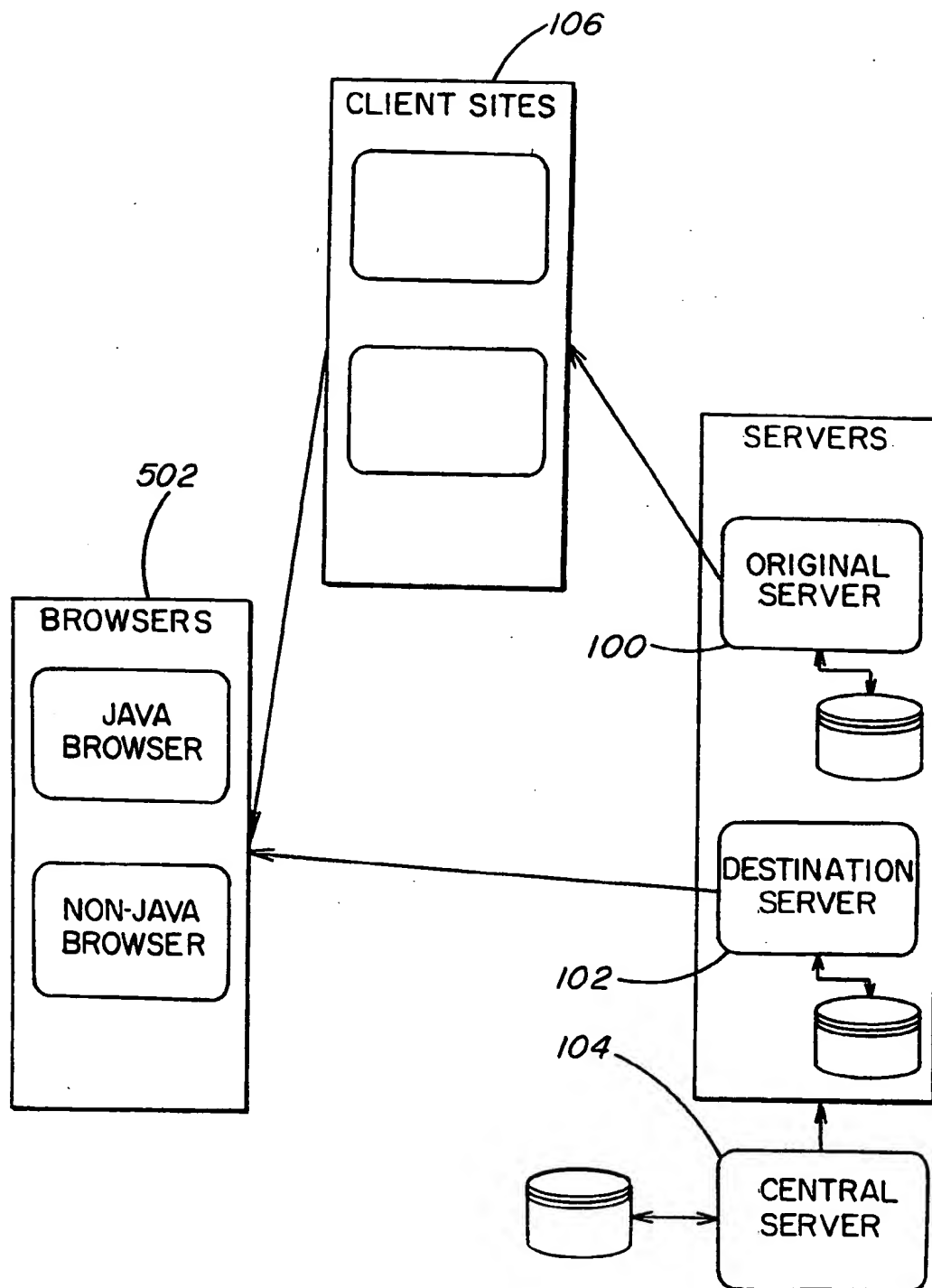


FIG. 5

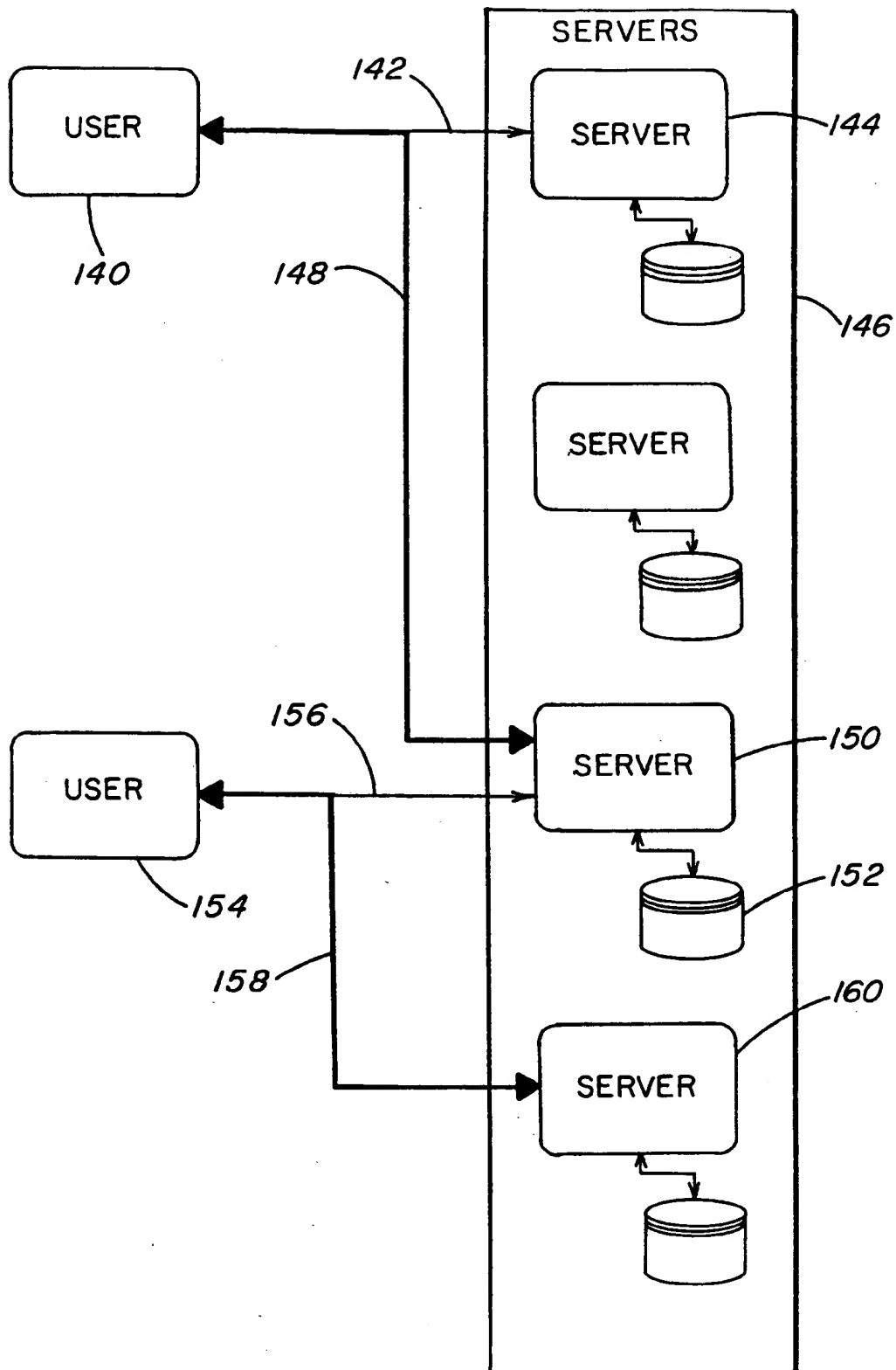
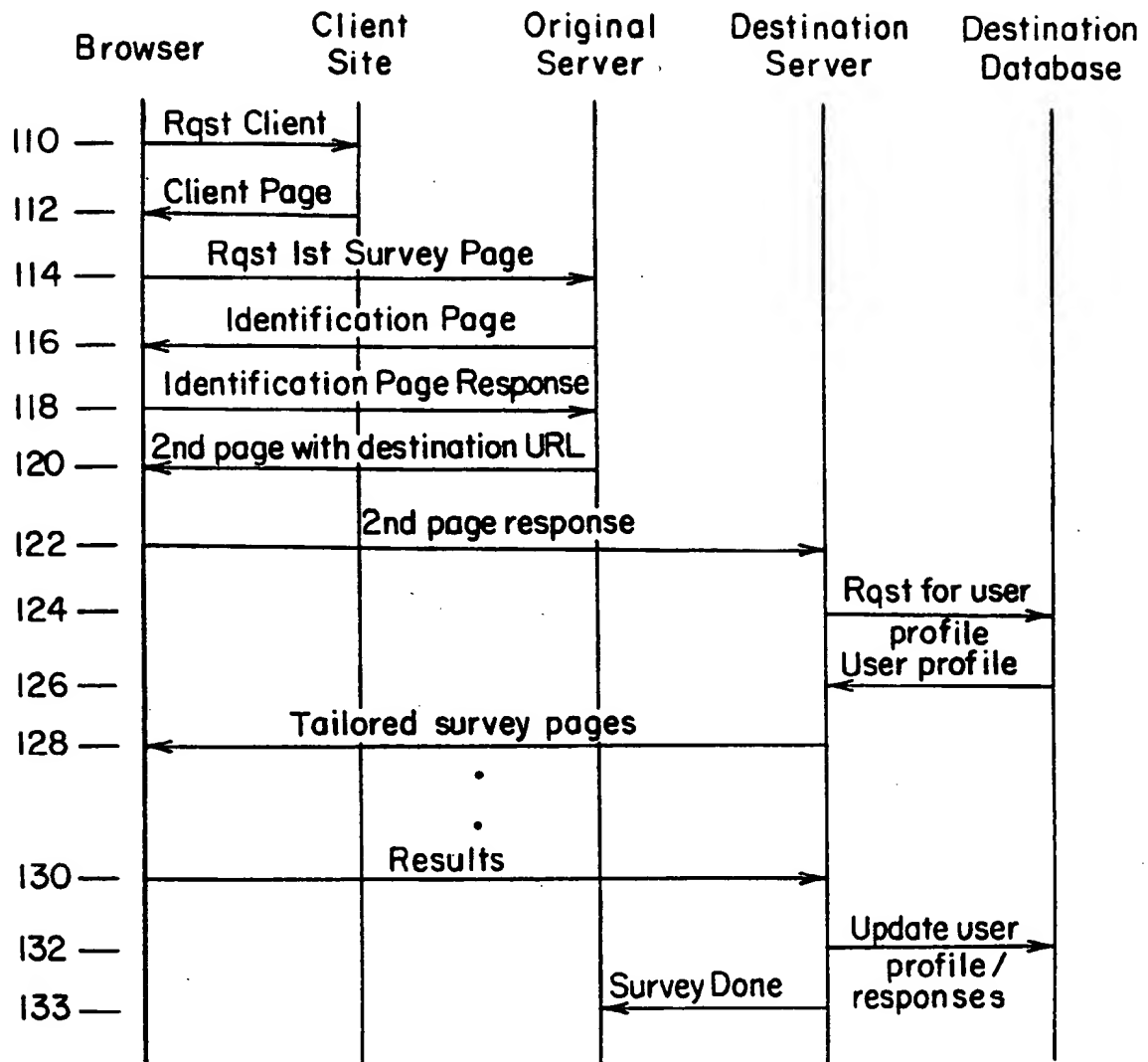
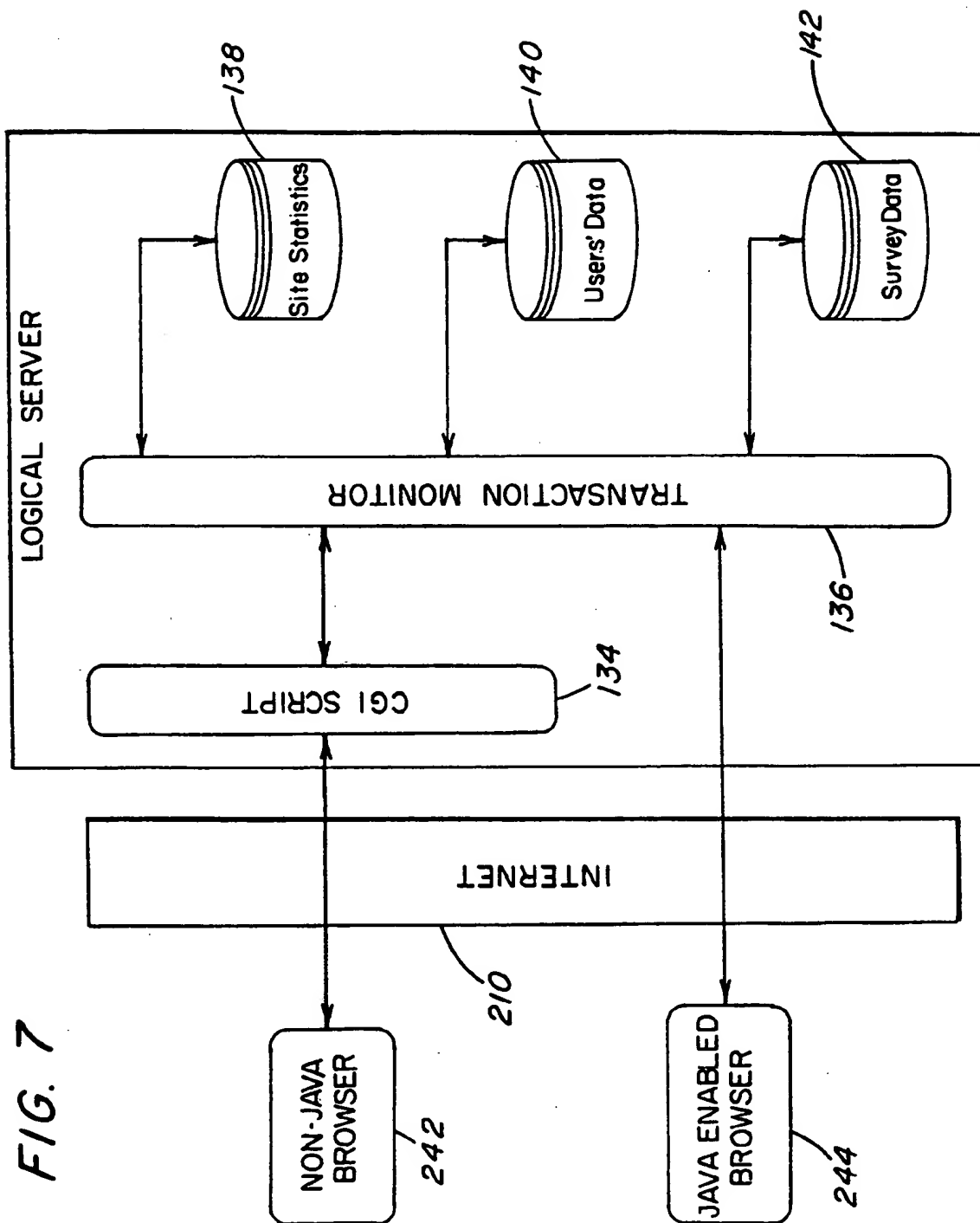


FIG. 6





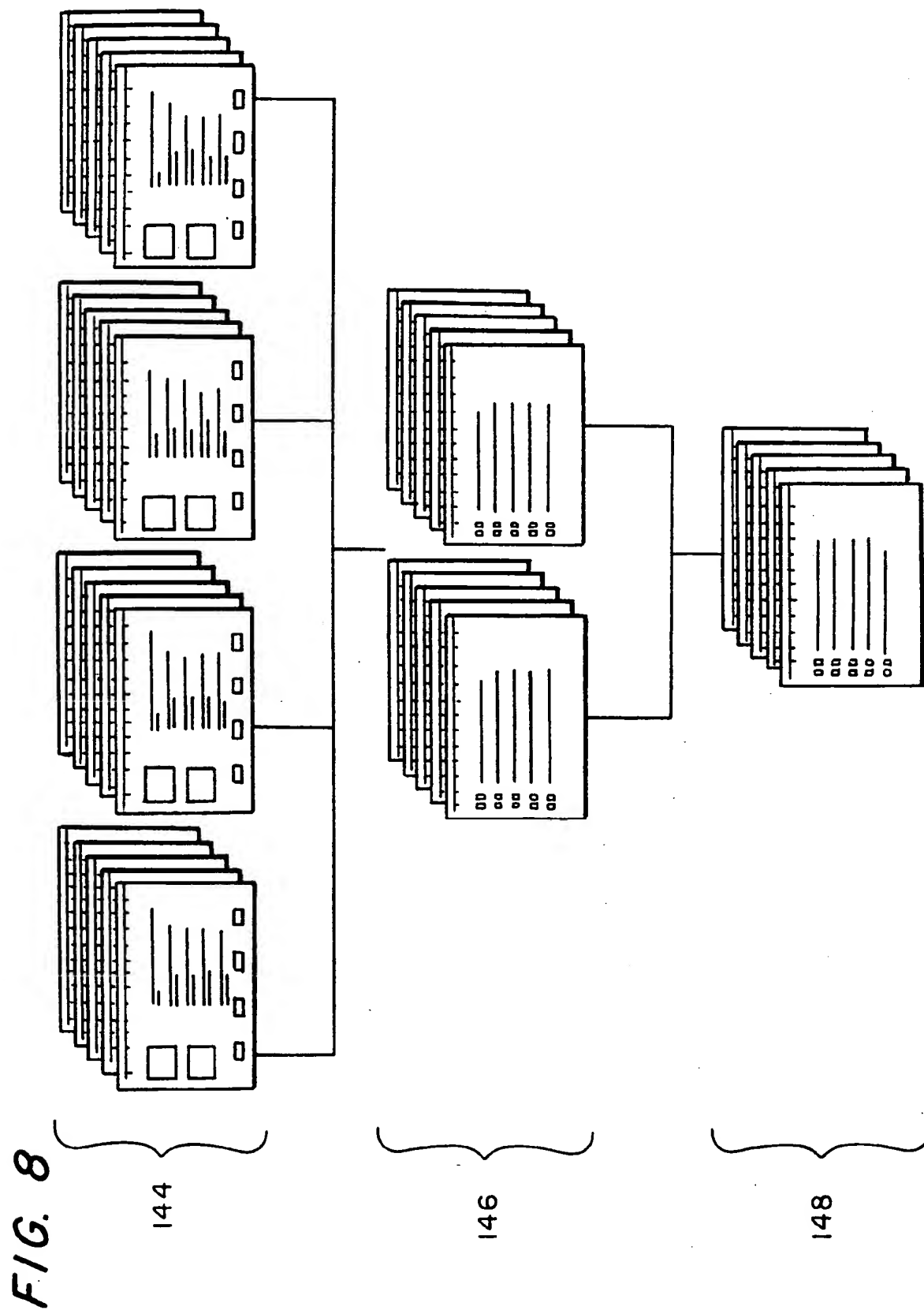
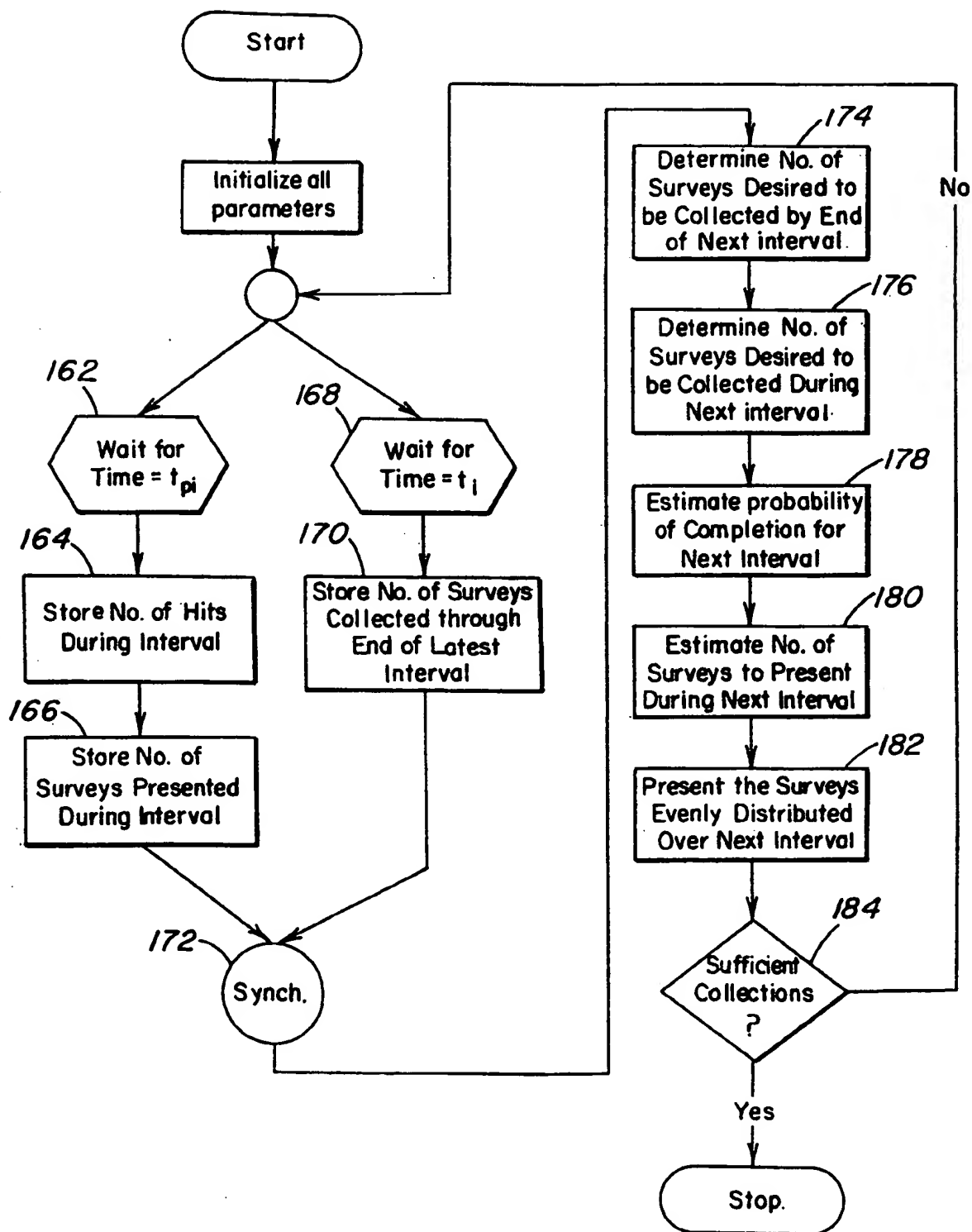
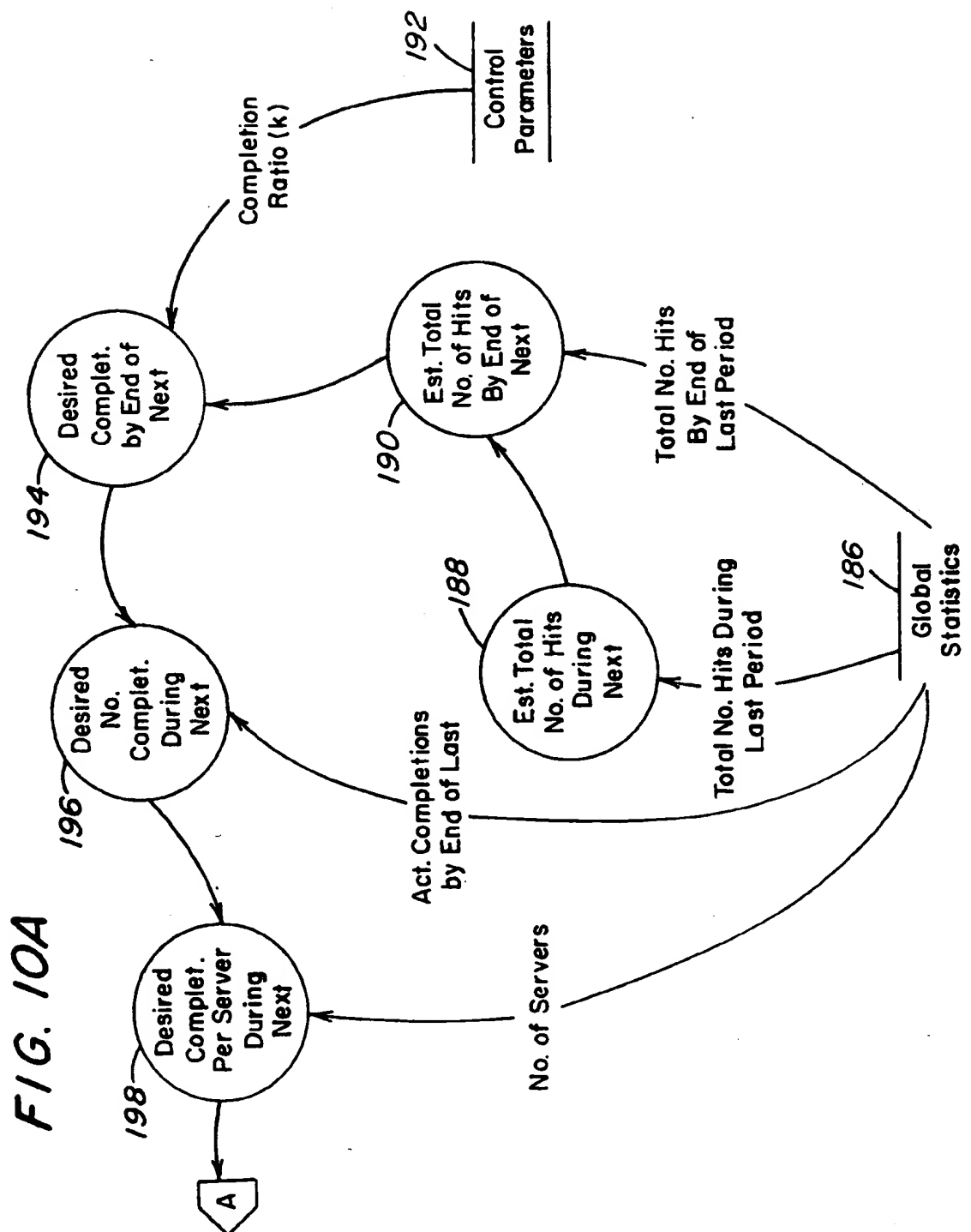
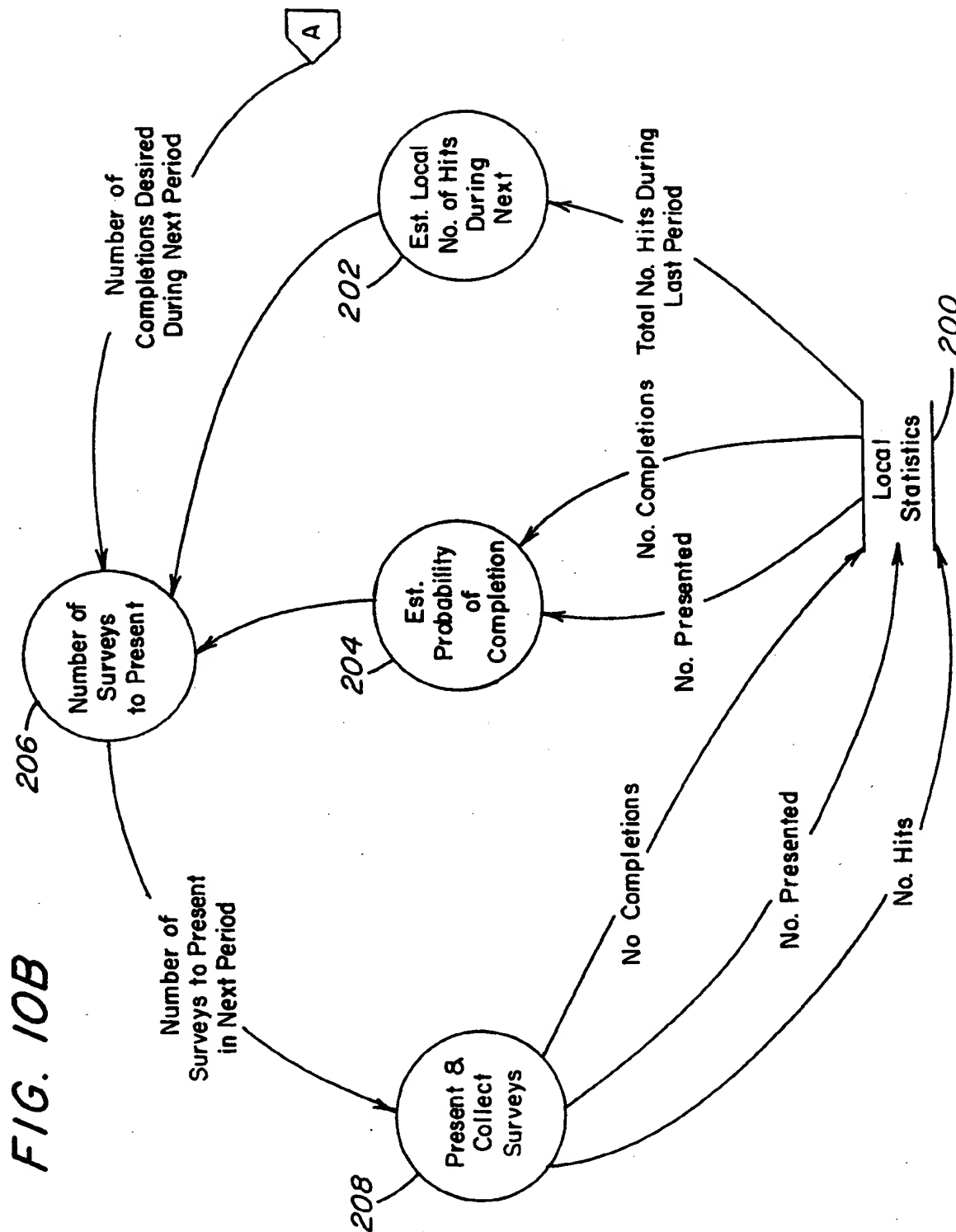


FIG. 9







12 / 24

```

import java.util.*;
import java.io.*;

//*****
// Class Tm - the Transaction Monitor
//*****
public class Tm {

// CONSTANTS
private int dtMinute=30;
private int tcMinute=2;
private double pc=0.1;
private double include=1;

// VARIABLES

// Default control parameters if none specified by clients
public int periodDays=7; // Number of days
public int cFinal=2000; // Number of surveys to collect
public int hFinal=10000; // Minimum number of hits

/* Number of hits, shown surveys and collected surveys */
public int h[];
public int s[];
public int c[];

// RING BUFFERS
private int nfit1=5; // Size of ring buffer 1
private int nfit2=4; // Size of ring buffer 2
private int nfit3=4; // Size of ring buffer 3

private int rpointer1; // Pointers to current location in
each of the ring buffers
private int rpointer2;
private int rpointer3;

private double[] ring1; // h, number of hits
private double[] ring2; // c, number of collections
private double[] ring3; // s, number shown

private final double a=-1.78; //Weights used in error calculation
for ds
private final double b=0.36;

/* Variables used to convert to seconds */
private int period;
private int dt;
private int T0; // Initial value for T

```

FIG. 11A

13 / 24

```
private double k; // Percentage of hits to be maintained

/* Values for a certain period */
private double hNext;
private double yNext;

/* Values used to calculate frequency */
private double dc;
private double dh;
private double ds;
private double old_dc;
private double dif_dc;
private double error;
private double old_error;

private double cperiod;
private double speriod;

// other variables
private int i=0;
private int bye=0;
private long my_time;
private long next_time;

// FILES: HOST SITE STATISTICS, SURVEYS AND DATA
// WE'LL ONLY USE HOST SITE STATISTICS

public File host_site;
public FileOutputStream file_writer3;
public BufferedOutputStream file_writer2;
public PrintStream file_writer;

/* MAIN LOOP, WAITING FOR APPLET'S TO CONTACT, AND
UPDATING VALUES EVERY 30 MINUTES */

public static void main(String args[]) {

    // temporary values for h,c and s (in every iteration).
    // To not have to write every time h[index]
    int temp_h;
    int temp_c;
    int temp_s;

    // These variables indicate the number of filled elements in
    the ring buffers.
    int full1=0;
```

FIG. 11B

```

int full2=0;
int full3=0;

/* FIRST INITIALIZATIONS */
Tm my_tm=new Tm();
my_tm.establish_values(0,2100,15000);

// INITIALIZE RING BUFFERS

my_tm.ring1=new double[my_tm.nfit1];    //Allocate storage for
the ring buffers
my_tm.ring2=new double[my_tm.nfit2];
my_tm.ring3=new double[my_tm.nfit3];

my_tm.rpointer1=my_tm.nfit1;            //Init all ring pointers
to last location in buffer
my_tm.rpointer2=my_tm.nfit2;
my_tm.rpointer3=my_tm.nfit3;

// Convert time periods to seconds
my_tm.period=my_tm.periodDays * 60 * 60 * 24;
my_tm.dt=my_tm.dtMinute * 60;
my_tm.T0=my_tm.tcMinute * 60;
my_tm.k= (double) (my_tm.cFinal/my_tm.hFinal);

/* Establishing values for a period */
my_tm.hNext= my_tm.hFinal * my_tm.dt / my_tm.period;
my_tm.yNext= (my_tm.cFinal * (my_tm.dt - my_tm.T0))/
my_tm.period;

// ESTABLISHING NEW DATE
Date my_date= new Date();
my_tm.my_time=my_date.getTime();
my_tm.next_time=my_tm.my_time+300000; //Interval of 30
minutes

/* Alternative: Implement a thread running background to
receive applet's requests
and update c[],h[] and s[] asynchronously */

/* Initializing for first sample period */
/* These arrays are defined for one year */
my_tm.h=new int[17520];
my_tm.s=new int[17520];
my_tm.c=new int[17520];

my_tm.h[0]=0;
my_tm.s[0]=0;

```

FIG. 11C

15 / 24

```
my_tm.c[0]=0;

Recv receive_thread= new Recv();
receive_thread.start();

// temporary
my_tm.make_files();

// MAIN LOOP
do {
    /* Get current time */
    Date next_date=new Date();
    my_tm.my_time=next_date.getTime();

    /* If 30 minutes are gone... */
    if (my_tm.my_time>=my_tm.next_time){

        /// Updating next time
        my_tm.next_time=my_tm.next_time+300000;

        // GET VALUES FOR CURRENT PERIOD
        temp_h=receive_thread.get_h();
        temp_c=receive_thread.get_c();
        temp_s=receive_thread.get_s();

        my_tm.pc=temp_c/temp_s;

        my_tm.h[my_tm.i]=temp_h;
        my_tm.c[my_tm.i]=temp_c;
        my_tm.s[my_tm.i]=temp_s;

        my_tm.error=my_tm.yNext-temp_c;

        /* CALL TO UPDATING */
        /* To update values to calculate frequency */

        my_tm.updating(my_tm.hNext,my_tm.yNext,my_tm.k,my_tm.error,my_
tm.incl,my_tm.pc);

        /* UPDATE VALUES IN HOST SITE FILES */

        my_tm.file_updating(my_tm.h[my_tm.i],my_tm.c[my_tm.i],my_tm.s[
my_tm.i],my_tm.i);

        /* Call contact other servers */
        my_tm.coherency();
```

FIG. 11D

```

    my_tm.i++;

    // RING BUFFERS !!! -> Update values for all 3 ring
buffers
    // WE'LL USE THE SAME FUNCTIONS TO WORK WITH RING BUFFERS

    // Ring buffer for h
    if (full1<my_tm.nfit1)
        full1++;

    my_tm.rpointer1=my_tm.new_ring_pointer(my_tm.rpointer1,my_tm.n
fit1);
    my_tm.ring1=my_tm.update_ring(my_tm.ring1, temp_h,
my_tm.rpointer1);

    // Ring buffer for pc
    if (full2<my_tm.nfit2)
        full2++;
    my_tm.rpointer2=my_tm.new_ring_pointer(my_tm.rpointer2,
my_tm.nfit2);
    my_tm.ring2=my_tm.update_ring(my_tm.ring2, temp_c,
my_tm.rpointer2);

    // Ring buffer for incl
    if (full3<my_tm.nfit3)
        full3++;
    my_tm.rpointer3=my_tm.new_ring_pointer(my_tm.rpointer3,
my_tm.nfit3);
    my_tm.ring3=my_tm.update_ring(my_tm.ring3, temp_s,
my_tm.rpointer3);

    // Reset local values for next period:
    my_tm.h[my_tm.i]=0;
    my_tm.c[my_tm.i]=0;
    my_tm.s[my_tm.i]=0;

    // Reset h,c,s values in the receive thread for the next
period
    receive_thread.set_h();
    receive_thread.set_c();
    receive_thread.set_s();

    // UPDATE YNEXT, PC, INCL

    // YNEXT
    // First, we update

```

FIG. 11E

17 / 24

```

    my_tm.hNext=my_tm.approxhNext(my_tm.hNext,my_tm.rpointer1,my_t
m.i,my_tm.ring1,full1,my_tm.nfit1);
    // LAST PARAMETER IS USED FOR THE 6 FIRST ITERATIONS,
    // WHEN THE RING BUFFER IS STILL NOT FULL !!!
    my_tm.yNext= my_tm.k * my_tm.hNext;

    // PC
    my_tm.pc=my_tm.approxProb(my_tm.pc, my_tm.rpointer2, my_tm.i,
my_tm.ring2, full2, my_tm.nfit2);
    // INCL
    my_tm.incl=my_tm.approxIncl(my_tm.incl, my_tm.rpointer3,
my_tm.i, my_tm.ring3, full3, my_tm.nfit3);

    }

    }while (c[i]<final); //end of MAIN LOOP
}

// CODE FOR RING BUFFER

// Update ring pointer to next location, moves backwards, at
zero
// it resets to end
int new_ring_pointer(int ring_pointer, int Size){

    if (ring_pointer<=0)
        ring_pointer=ring_pointer + Size - 1;
    else
        ring_pointer--;

    return ring_pointer;
}

//Insert data into next location in ring
double[] update_ring(double[] RingArr, double Val, int
ring_pointer){

    RingArr[ring_pointer]=Val;

    return RingArr;
}

// CODE TO CALCULATE NEXT HNEXT
double approxhNext(double h, int ring_pointer, int I, double[]
h_ring, int full1, int tamany){

```

FIG. 11F


```

double[] c;
double hacu=0;

c=new double[5];    //Weights to be used for estimation
c[0]=0.3;
c[1]=0.25;
c[2]=0.20;
c[3]=0.15;
c[4]=0.1;

for (int j=0; j<full1; j++)
    hacu=hacu + (c[j] * h_ring[tamany-j-1]);
hacu=hacu/full1;

return hacu;
}

// CODE TO CALCULATE NEXT PC
double approxProb(double pc,int ring_pointer,int i,double[]
ring,int full2,int tamany){

    double[] c;
    double pcacu=0;

    c=new double[4];    //Weights to be used for estimation
    c[0]=0.35;
    c[1]=0.30;
    c[2]=0.20;
    c[3]=0.15;

    for (int j=0;j<full2;j++){
        pcacu=pcacu+(c[j]*ring[tamany-j-1]);
        file_writer.println("--> "+ring[tamany-j-1]);
    }
    pcacu=pcacu/full2;
    file_writer.print("PCACU:#+pcacu+"#");

    return pcacu;
}

// CODE TO CALCULATE NEXT INCLINATION (SLOPE)

double approxIncl(double incl,int ring_pointer,int i,double[]
ring,int full3,int tamany){

    double[] c;
    double inclacu=0;

```

FIG. 11G

19 / 24

```

c=new double[4];    //Weights to be used for estimation
c[0]=0.35;
c[1]=0.30;
c[2]=0.20;
c[3]=0.15;

for (int j=0;j<full3;j++)
    inclacu=inclacu+(c[j]*ring[tamany-j-1]);
inclacu=inclacu/full3;

return inclacu;
}

public void updating(double hNext,double yNext,double k,double
error,double incl,double pc){

    /* MAIN LOOP */
    if (i!=0)
        old_dc=dc;
    else
        old_dc=yNext-c[i];    // First value

    dc = yNext - c[i];        // Number to collect in next period
    dh = hNext-h[i];          // Number of hits in next period
    dif_dc = dc - old_dc;    //Unused?

    if (i==0) {
        if (pc>0.0)
            //ds is lesser of expected hits or dc/pc
            ds=Math.min(dh, (dc/pc));
        else
            ds=dh;
    }
    else {    // i!=0
        if (pc>0.0)
            // ds is lesser of expected hits or previous ds corrected
            // for error
            ds=Math.min(dh, (((ds+dc/pc)/2) + ((error * a)*(1-
(b*incl))))));
        else
            ds=dh;
    }
}

// INTERFACE FUNCTION TO CHANGE CONTROL PARAMETERS
// TO BE USED BY CLIENTS, BEFORE STARTING SURVEY

// To allow for changing only certain values:

```

FIG. 11H

20 / 24

```
// If a parameter=0 -> this value will remain unchanged

public void establish_values(int Duration, int Collections, int
Hits) {

    if (Duration!=0)
        periodDays=Duration;
    if (Collections!=0)
        cFinal=Collections;
    if (Hits!=0)
        hFinal=Hits;
}

// INTERFACE FUNCTION -> This function creates the three files
// in the server (if they didn't exist)

public void make_files() {

try{
    host_site= new File("hostsite");
    file_writer3= new FileOutputStream(host_site);
    file_writer2= new BufferedOutputStream(file_writer3);
    file_writer= new PrintStream(file_writer2,true);

} catch(IOException e) {
    System.out.println(" PROBLEM CREATING FILES");
}

}

/// FUNCTION TO UPDATE FILE HOST SITE STATISTICS

public void file_updating(double h,double c,double s,int i){

try{
    file_writer.print(i+" "+h+" "+c+" "+s+"\n");
} catch (IOException e){
    System.out.println("PROBLEM WRITING TO HOST SITE STATISTICS");
}

}

// FUNCTION TO MANTAIN COHERENCY BETWEEN SERVERS !!!!

public void coherency(){

// Currently, being implemented.
}
```

FIG. 11I

```
} // End of class Tm

/* CODE WAITING TO RECEIVE APPLET'S REQUESTS, AS WELL AS
WHEN A SURVEY IS SHOWN OR FINISHED */

//*****
// Class Recv - thread for receiving statistics
//*****
class Recv extends Thread{

// We define temporary variables, h,s,c
// Number of hits, shown and collected surveys

public int h;
public int s;
public int c;

public int exit=0;

// INTERFACE: Methods to access these variables

public int get_h(){
    return h;
}

public int get_s(){
    return s;
}

public int get_c(){
    return c;
}

// Methods to reset variables

public void set_h(){
    h=0;
}

public void set_c(){
    c=0;
}

public void set_s(){
    s=0;
}
```

FIG. 11J

```
public void run(){  
    do(  
        yield();  
        // Code not yet implemented.  
        // Will receive data, identify the value, and update the  
        corresponding variable  
        )while(exit!=1);  
    }  
  
} //End of class Recv
```

FIG. 12

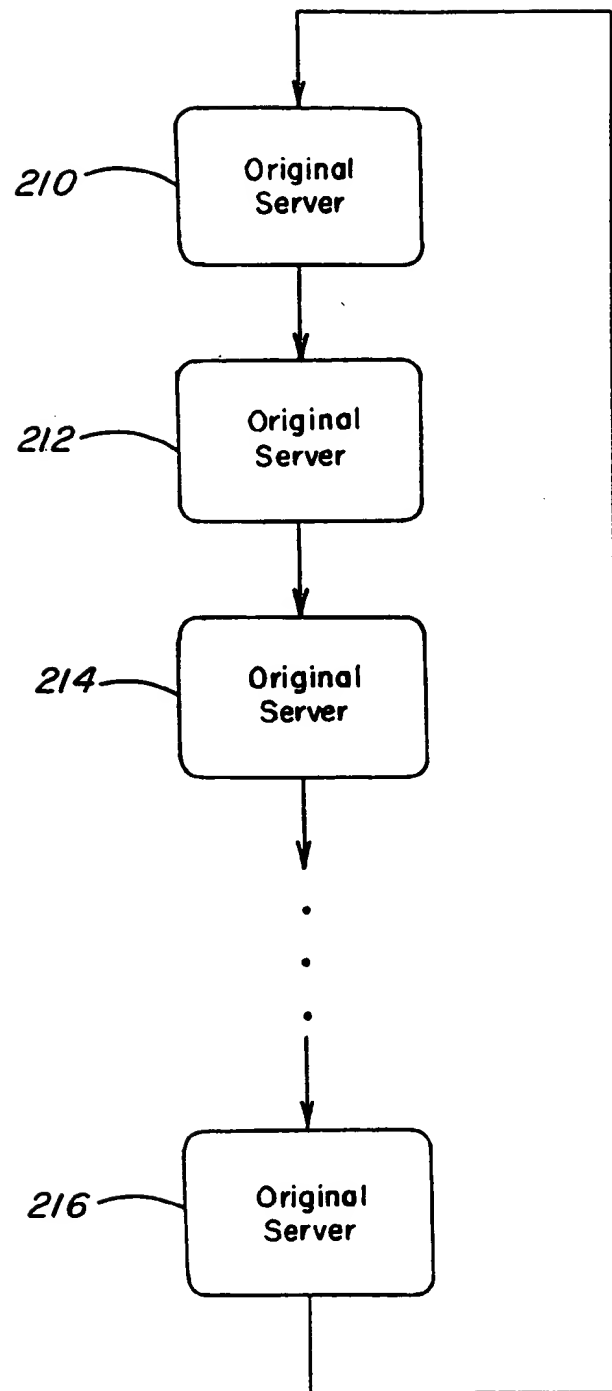


FIG. 13